

these records reach the mass storage device 111, further analysis, review, and subsequent

92 processing may be performed by the application developer as needed. --

In the Claims

Please amend claims 1, 13, and 18 as shown below:

1. (ONCE AMENDED) A computing system having a mass storage device and a system timer for obtaining benchmark timing for a portion of an application program execution, the application program having permanently inserted performance markers, the computing system comprising:

a mass storage system;

an init module for determining if the timestamp data is to be collected during the operation of the application program;

93 a performance marker module for obtaining and storing the timestamp data for later retrieval at predefined points corresponding to the permanently inserted performance markers;

an uninit module for formatting and storing the obtained timestamp data into a data file within the mass storage device that permits retrieval after the termination of the application program; and

a performance benchmark data post processing module for determining the benchmark timing from two or more timestamp data entries;

wherein

the init module is executed before any timestamp data is collected;

the performance marker module is executed each time benchmark timestamp data and overhead timestamp data is to be collected;

the uninit module is executed after all timestamp data desired has been collected; and

the performance benchmark data post processing module determines the benchmark timing from timestamp entries stored within the data file.

13. (ONCE AMENDED) A method for obtaining benchmark timing for a portion of an application program execution, the method comprising:

permanently inserting one or more code markers into the application program at locations within the application program corresponding to the point at which benchmark timing data is desired;

determining if benchmark timing data is to be collected at each code marker by checking for the existence of processing modules identified by an identification key within a system registry;

if benchmark timing data is to be collected at each code marker:

generating a benchmark data record containing the collected benchmark timing data each time the code markers are reached;

storing the benchmark data records within a data memory block within the processing modules identified by the identification key within the system registry;

retrieving the benchmark data records from the data memory block for transfer to a mass storage device once all of the run-time internal state data has been collected; and

Qy
and processing the benchmark data records stored within the mass storage device to determine the benchmark timing defined between two benchmark data records.

95 18. (ONCE AMENDED) A computer data product readable by a computing system and encoding a computer program of instructions for executing a computer process for obtaining run-time internal state data within an application program, said computer process comprising the steps of:

permanently inserting one or more code markers into the application program at locations within the application program corresponding to the point at which benchmark timing data is desired;

Determining if benchmark timing data is to be collected at each code marker by checking for the existence of processing modules identified by an identification key within a system registry;

if benchmark timing data is to be collected at each code marker:

generating a benchmark data record containing the collected benchmark timing data each time the code markers are reached;

storing the benchmark data records within a data memory block within the processing modules identified by the identification key within the system registry;

retrieving the benchmark data records from the data memory block for transfer to a mass storage device once all of the run-time internal state data has been collected; and

processing the benchmark data records stored within the mass storage device to determine the benchmark timing defined between two benchmark data records.
